

IN THE CLAIMS:

Please amend the claims as indicated below.

1. (Previously Presented) A method for compressing a Rabin signature, s, for a user
5 having a public key, n, comprising the step of:

generating a compressed Rabin signature based on a continued fraction expansion
of s/n, wherein said continued fraction expansion of s/n further comprises the steps of

- 10 computing principal convergents, u_i/v_i , for i equal to 1 to k, of a continued
fraction expansion of s/n, where k is a largest integer for which principal convergents are
defined;

establishing an index l , such that $v_l < \sqrt{n} < v_{l+1}$; and

generating a compressed Rabin signature (v_l, m) for a message, m.

2. (Cancelled)

- 15 3. (Original) A method for compressing a Rabin signature, s, for a message, m, and a
user having a public key, n, comprising the steps of:

computing principal convergents, u_i/v_i , of a continued fraction expansion of s/n;

establishing an index l , such that $v_l < \sqrt{n} \leq v_{l+1}$; and

- 20 generating a compressed Rabin signature (v_l, m) .

4. (Original) The method according to claim 3, wherein $sv=u \pmod n$.

- 25 5. (Original) The method according to claim 3, wherein $|v| \leq \sqrt{n}$.

6. (Original) The method according to claim 3, wherein $|u| \leq \sqrt{n}$.

- 30 7. (Original) The method according to claim 1, wherein said principal convergents,
 u_i/v_i , are computer for i equal to 1 to k, where k is a largest integer for which principal
convergents are defined.

8. (Original) A method for decompressing a compressed Rabin signature (v, m) for a message, m, and user having a public key, n, comprising the steps of:
5 applying a message formatting function, h, to the message, m, to computing $h(m)$;
 computing a value, t, as $h(m)v^2 \bmod n$;
 obtaining a value, w, as a square root of the value, t;
 computing a signature value, s, as $w/v \bmod n$; and
 providing a decompressed signature (s,m).
9. (Original) The method of claim 8, further comprising the step of generating an error if no integer square root exists.
10. (Original) A method for compressing an RSA signature, s, for a message, m, and a user having a public key (n, e), comprising the steps of:
15 computing principal convergents, u_i/v_i , of the continued fraction expansion of s/n ;
 establishing an index l, such that $v_l < n^{(1-1/e)} \leq v_{l+1}$; and
 generating a compressed signature (v_l, m).
11. (Original) A method for decompressing a RSA signature (v, m) for a message, m, and a user having a public key (n, e), comprising the steps of:
20 applying a message formatting function, h, to the message, m, to computing $h(m)$;
 computing a value, t, as $h(m)v^e \bmod n$;
 determining whether the values t or $t-n$ have an e^{th} root over integer values;
 computing a value, w, as the e^{th} root; and
 computing the decompressed signature ($w/v \bmod n, m$).
25
12. (Original) The method of claim 11, further comprising the step of generating an error if no e^{th} root exists.
13. (Previously Presented) A system for compressing a Rabin signature, s, for a user having a public key, n, comprising:
30 a memory; and

at least one processor, coupled to the memory, operative to:

generate a compressed Rabin signature based on a continued fraction expansion of s/n, wherein said processor is further configured to perform said continued fraction expansion of s/n by:

5 computing principal convergents, u_i/v_i , for i equal to 1 to k, of a continued fraction expansion of s/n, where k is a largest integer for which principal convergents are defined;

establishing an index l , such that $v_l < \sqrt{n} < v_{l+1}$; and

generating a compressed Rabin signature (v_l, m) for a message, m.

10

14. (Cancelled)

15. (Original) A system for decompressing a compressed Rabin signature (v, m) for a message, m, and user having a public key, n, comprising:

15 a memory; and

at least one processor, coupled to the memory, operative to:

apply a message formatting function, h, to the message, m, to computing $h(m)$;

compute a value, t, as $h(m)v^2 \bmod n$;

obtain a value, w, as a square root of the value, t;

20 compute a signature value, s, as $w/v \bmod n$; and

providing a decompressed signature (s,m) .

16. (Original) The system of claim 15, wherein said processor is further configured to generate an error if no integer square root exists.

25